REMARKS

Claims 1-30 remain herein.

This Preliminary Amendment is submitted to correct clerical errors which occurred during translation of the Japanese language text into the English language.

Examination of this application on its merits is respectfully requested.

Respectfully submitted,

PARKHURST & WENDEL, L.L.P.

September 14, 2001
Date

Roger W. Parkhurst
Registration No. 25,177

Attachment:
Specification Mark Up

RWP/ame

Attorney Docket No. HYAE:127

PARKHURST & WENDEL, L.L.P.
1421 Prince Street, Suite 210
Alexandria, Virginia 22314-2805
Telephone: (703) 739-0220

required for the above-described writing and reading.

The address generation unit shown in figure 1 sequentially generates addresses of the storage unit 104 by executing the address generation rule defined by formula (3).

That is, in the address generation unit 103, utilizing that "(X+Y)modZ = XmodZ+YmodZ" holds, calculation of the (b-x)th power of M in the term "α×M**(b-x)mod(L×M-1)" in "(Ab(n-1)+α×M**(b-x))mod(L×M-1)" in formula (3) is executed by repeating multiplication of M using the constant generator 110, the multiplier 111, and the register 113, and multiplication of α in this term and remainder calculation by (L×M-1) are executed using the overflow processing unit 140.

Further, calculation of the term "Ab(n-1)mod(L×M-1)" in formula (3) and input of the initial value Ab(0)=0 are executed by the overflow processing unit 141.

Further, addition of the results of the remainder calculations in these two terms is executed by the adder 115.

The selector 121 receives the ~~input~~ output of the overflow processing unit 140 and the output of the selector 124. When the input data corresponds to the head of the block, a block head input data sync signal 102 is input, and the selector 121 selects the output of the multiplier 111. In other cases, the selector 121 selects the output of the selector 124. The output of the selector 121 is compared with L×M-1 by the comparator 123. The selector 124 receives the output of the subtracter 122 which

subtracts L×M-1 from the output of the selector 121, and the output of the selector 121. When the comparator 123 decides that the output of the selector 121 is equal to or larger than L×M-1, the selector 124 selects the output of the subtracter 122. In other cases, the selector 124 selects the output of the selector 121. The output of the selector 124 is inputted to the register 113. In this way, when the input to the overflow processing unit 140 exceeds L×M-1, the overflow processing unit 140 repeats subtraction of L×M-1 from the input to keep the value equal to or smaller than L×M-1.

The overflow processing unit 140 prevents the numerical values from diverging over L×M-1 due to repetition of multiplication or addition in the address generation unit 103.

In the address generation unit 103 shown in figure 1, the constant generator 118 generates an initial value "α" and outputs it to the register 113. The multiplier 111 multiplies the output of the register 113 by the output "M" from the constant generator 110 and outputs the product to the overflow processing unit 140.

When the input data to the overflow processing unit 140 exceeds L×M-1, the overflow processing unit 140 repeats subtraction of "L×M-1" by an internal loop until the input data becomes equal to or smaller than L×M-1, and outputs the result to the register 113. The output of the register 113 is again multiplied by the output "M" of the constant generator 110 by the multiplier 111, and the product is inputted to the overflow

output value "0" from the register 117, and the selectors 134 and 130 select this sum "2" and input it to the register 117.

Since the output value from the register 117 is "0", by using this as an address of the storage unit 104, an initial value (indefinite value) is read from the storage unit 104 at the timing of "H" of a control signal (write enable signal) NWE, and the data D0 which has been retained in the register 129 from time t3 is input to the storage unit 104 at the timing of "L" of the control signal (write enable signal) NWE. Although these states are identical on and after time t5, since the selector 130 selects the output of the selector 134 and the output of the register 127 holds the value "2", the output from the adder 115 increments by "2" every time one CLK signal is input. However, when the output from the adder 115 comes to be larger than "19", the selector 134 selects the output from the subtracter 132 to suppress the value to "19" or smaller.

At time t23, when the selector 121 selects the output value "50" from the multiplier 111, the selector 124 selects the output of the subtracter ~~123~~ 122 according to the decision of the comparator 123 and outputs a value "31" (= 50-19). The selector 126 selects this value and inputs it to the register 113. Further, the selector 128 selects the output of the register 113 and inputs its value "10" to the register 127.

The adder 115 adds the output value "2" from the register 127 and the output value "19" from the register 117. At time t23,

the selector 119 selects not the output from the adder 115 but the output value "0" from the constant generator 119, and inputs it to the register 117.

The addresses shown in figure 2(a) are generated by the above-described operation from time t4 to time t23. Further, the initial value (indefinite value) is sequentially read from these addresses of the storage unit 104 every time a clock CLK is inputted, and the data D0 to D19 are sequentially written in these addresses at every input of clock CLK.

At time t24, the register 113 outputs the value "31" while the multiplier 111 outputs the value "155", and the selector 121 selects the output value "31" from the selector 124. The selector 124 selects the output value "12" from the subtracter 122 according to the decision of the comparator 123, and the selector 126 inputs this value "12" to the register 113.

Since the selector 128 inputs the output value "10" from the selector ~~127~~ 128 to the ~~selector~~ register 127, this value "10" is retained.

Further, the adder 115 adds the output value "10" from the register 127 and the output value "0" from the register 117, and the selector 134 selects the sum "10" according to the decision of the comparator 133 and inputs it to the register 117.

At time t25, the register 113 outputs the value "12" while the multiplier 111 outputs the value "60", and the selector 121 selects the output value "12" from the selector 124. The selector 126 inputs this value "12" to the register 113.

generating operation of the address generation unit 3, required for performing the above-described writing and reading.

The address generation unit shown in figure 7 sequentially generates addresses of the storage unit 4 by executing the address generation rule defined by formula (4).

That is, in the address generation unit shown in figure 7, by utilizing that "(X+Y)modZ = XmodZ+YmodZ" holds, calculation of the (b-x)th power of L in the term "α×L**(b-x)mod(L×M-1)" in "(Ab(n-1)+α×L**(b-x))mod(L×M-1)" of formula (4) is executed by repeating multiplication of L by using the constant generator 10, the multiplier 11, and the register 13, and further, multiplication of α and remainder calculation by (L×M-1) in this term are executed by using the overflow processing unit 40.

Further, calculation of the term "Ab(n-1)mod(L×M-1)" in formula (4) and inputting of the initial value Ab(0)=0 are executed by the overflow processing unit 41.

Further, addition of results of remainder calculations in these two terms is executed by the adder 15.

The selector 21 is given the ~~input~~ output of the overflow processing unit 40 (output of the multiplier 11) and the output of the selector 24. When the input data corresponds to the head of the block and the head input data sync signal 2 is inputted, the selector 21 selects the output of the multiplier 11. In other cases, the selector 21 selects the output of the selector 24. The output of the selector 21 is compared with L×M-1 by the

comparator 23. The selector 24 receives the output of the subtracter 22 which subtracts L×M-1 from the output of the selector 21, and the output of the selector 21. When the comparator 23 decides that the output of the selector 21 is equal to or larger than L×M-1, the selector 24 selects the output of the subtracter 22. In other cases, the selector 24 selects the output of the selector 21. The output of the selector 24 is inputted to the register 13. In this way, when the input to the overflow processing unit 40 exceeds L×M-1, the overflow processing unit 40 repeats subtraction of L×M-1 from the input to make the input value equal to or smaller than L×M-1.

The overflow processing unit 40 prevents the numerical value from diverging over L×M-1 due to repetition of multiplication and addition in the address generation unit 3.

In the address generation unit 3 shown in figure 7, the constant generator 18 generates an initial value "α" and outputs this to the register 13. The multiplier 11 multiplies the output of the register 13 by the output "L" of the constant generator 10 and outputs the product to the overflow processing unit 40.

When the input data to the overflow processing unit 40 exceeds L×M-1, the overflow processing unit 40 repeats subtraction of "L×M-1" by an internal loop until the input data becomes equal to or smaller than L×M-1, and outputs the result to the register 13. The output of the register 13 is again multiplied by the output "L" of the constant generator 10 by the

inputs it to the register 17.

The addresses shown in figure 8(a) are generated by the above-described operation from time t4 to time t23. Further, the initial value (indefinite value) is sequentially read from these addresses of the storage unit 4 every time one clock CLK is inputted, and the data D0 to D19 are sequentially written in these addresses at every clock CLK input.

At time t24, the register 13 outputs the value "13" while the multiplier 11 outputs the value "52", and the selector 21 selects the output value "13" from the selector 24. The selector 24 selects the output value "13" from the selector 21 according to the decision of the comparator 23, and the selector 26 inputs this value "13" to the register 13.

Since the selector 28 inputs the output value "13" from the selector ~~27~~ 28 to the ~~selector~~ register 27, this value "13" is retained.

Although these states are identical on and after time t25, since the selector 30 selects the output of the selector 34 and the output of the register 27 holds the value "8", the output of the adder 15 increases by "10" every time one CLK signal is inputted. However, when the output of the adder 15 comes to be larger than "19", the selector 34 selects the output of the subtracter 32 to suppress the value at "19" or smaller, and this is given as an address to the storage unit 4 after one clock CLK through the register 17.

Therefore, the addresses shown in figure 8(b) are generated